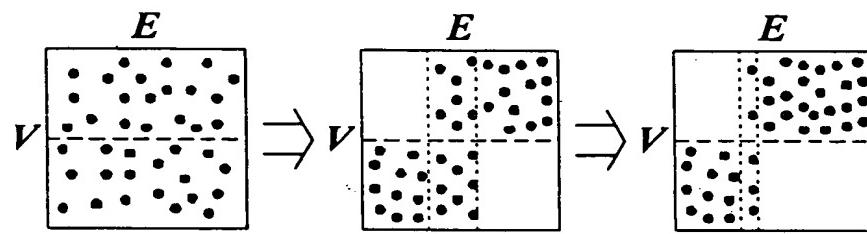


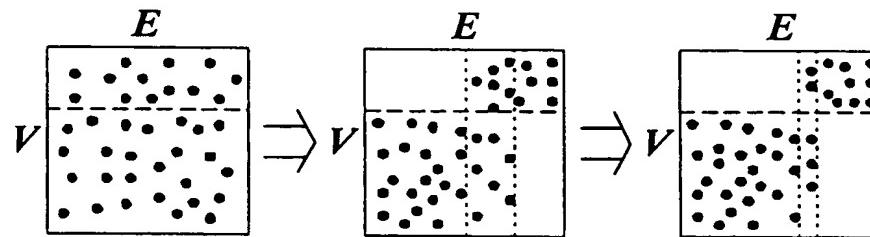
**Replacement Sheet**



APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	DRAFTSMAN

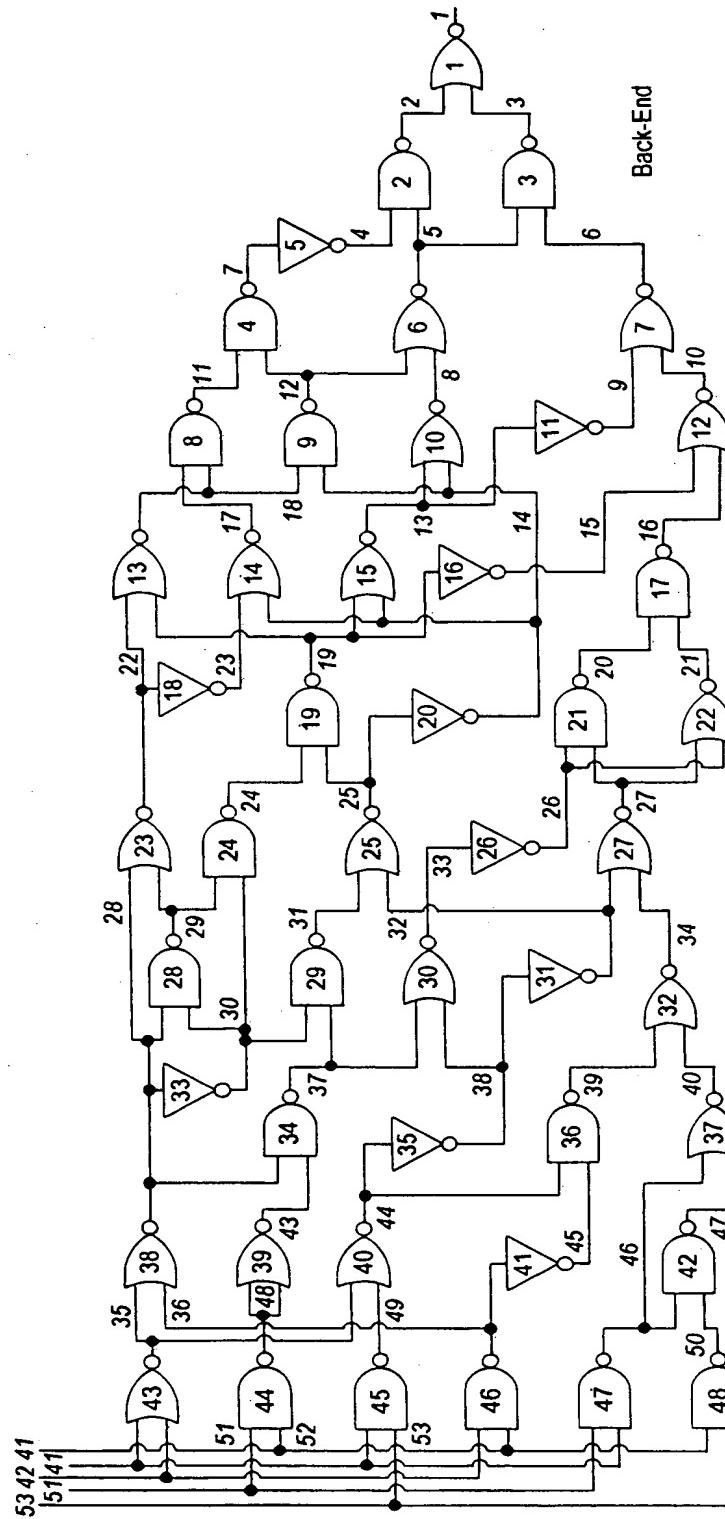
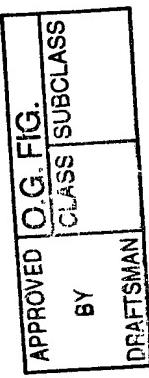


**FIG. 1**



**FIG. 2**

# Replacement Sheet



Front-End

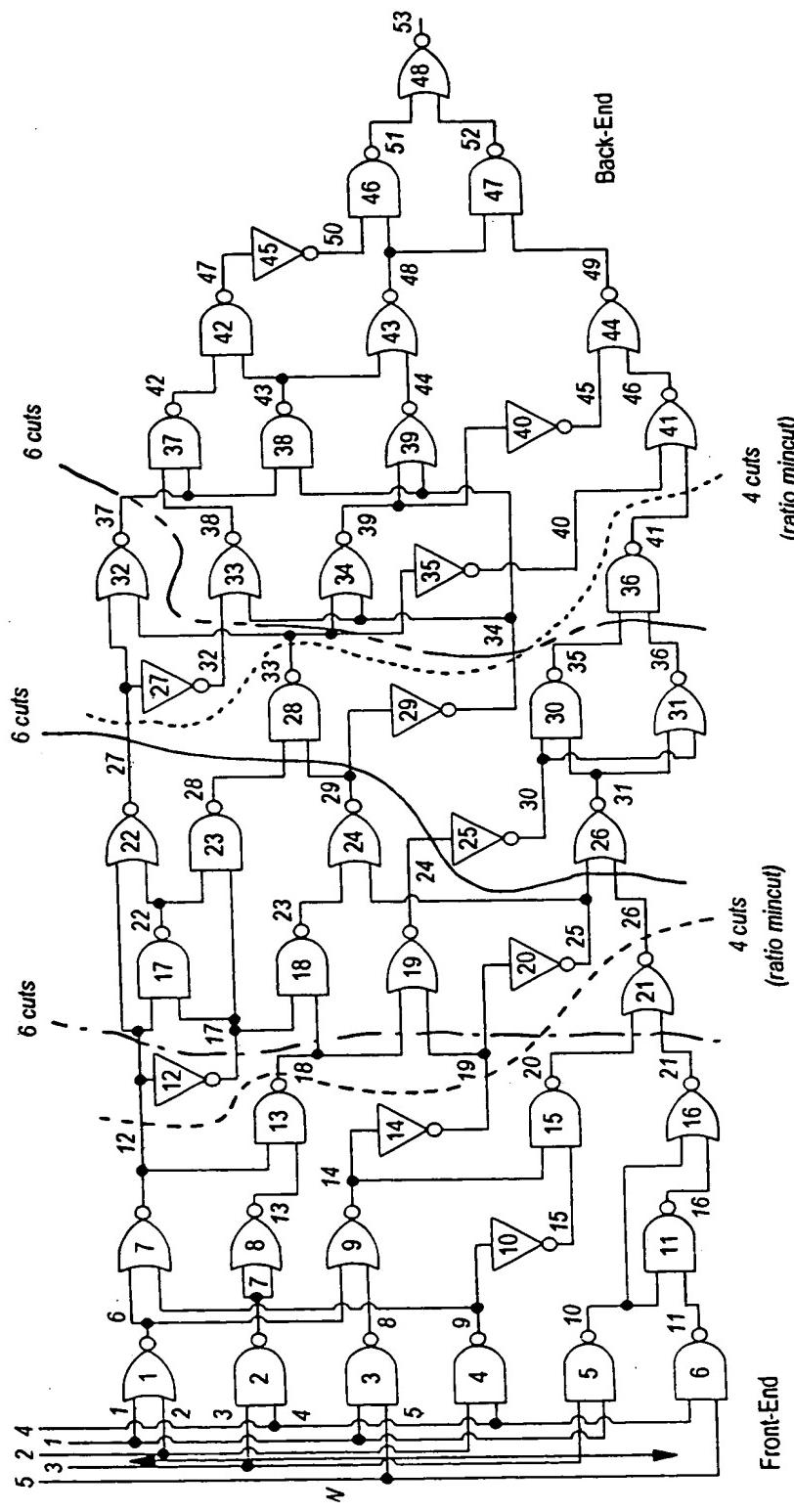
**FIG. 3**

# Replacement Sheet

APPROVED	O.G. FIG.
BY	CLASS
DRAFTSMAN	SUBCLASS

O I P E

SEP 02 2003

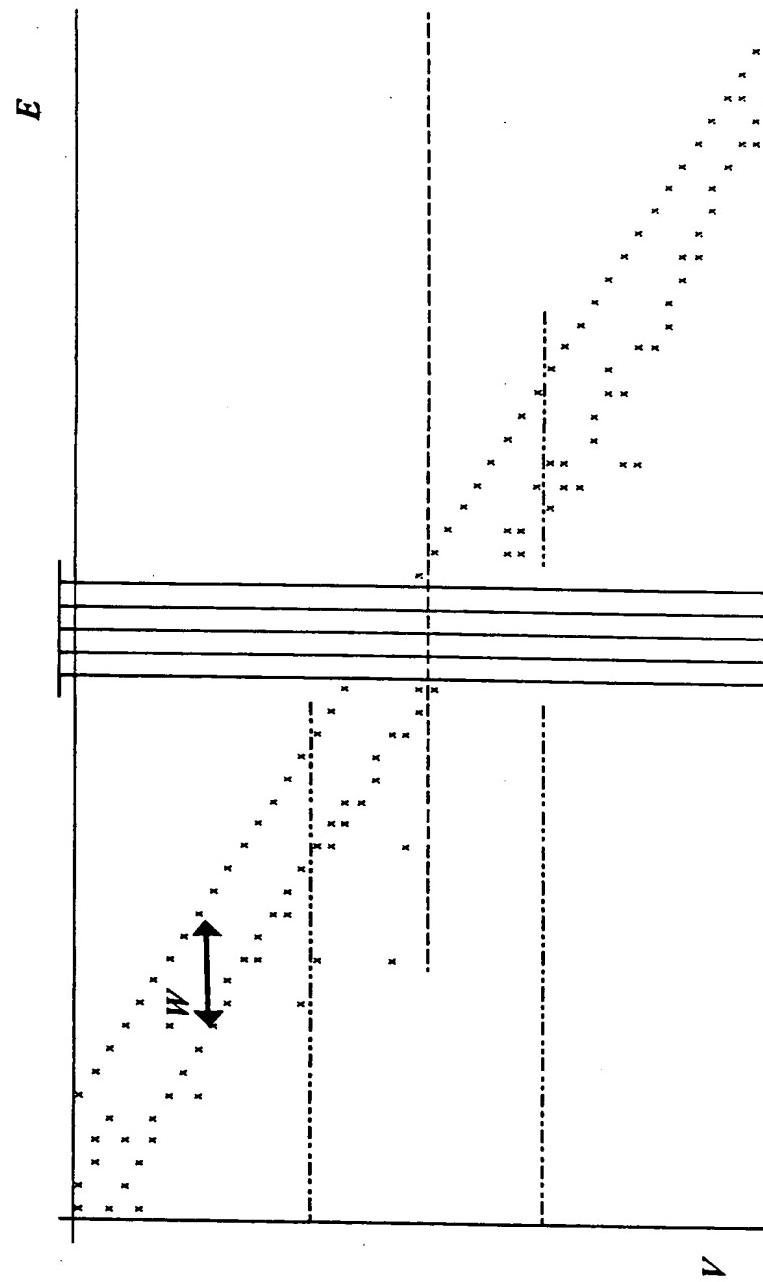


**FIG. 4**

**Replacement Sheet**

SEP 02 2003

APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	DRAFTSMAN

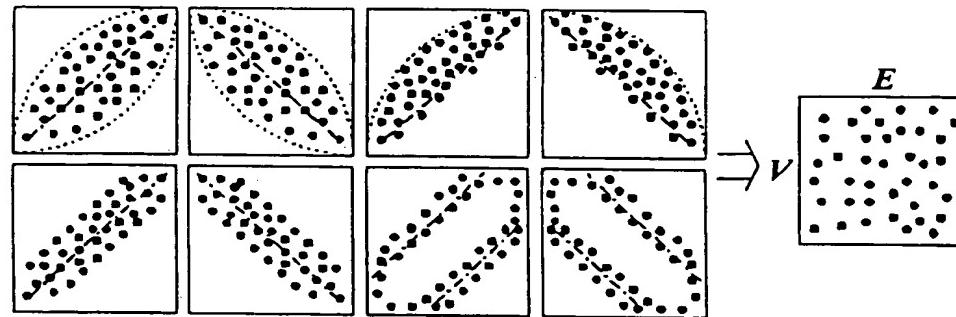


**FIG. 5**

# Replacement Sheet



APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	
DRAFTSMAN	



**FIG. 6**

# Replacement Sheet

APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	
DRAFTSMAN	

```

#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define Required_Num 48
int A[Required_Num], B[Required_Num], C[Required_Num];

int main(void)
{
    int i, j, m, n, seed, non_used;
    time_t t;

    for(i=0; i< Required_Num; i++)
    { A[i] = 0; B[i] = i+1; } /* For initialize */

    seed = (unsigned) time(&t); /* srand((unsigned) time(&t)); */
    srand( seed );

    printf("\nseed %u, random numbers from 1 to %d\n", seed, Required_Num);
    for(i= Required_Num-1; i>=0; i--)
    {
        int k;
        k = (rand() % Required_Num);
        printf("%2d\t", k+1);
        if( B[k] != 0 ) { A[i] = k+1; B[k] = 0; }

    }
    printf("\nArray A... Non-repeated generated numbers (from back-end):\n");
    for(i=0; i< Required_Num; i++) printf("%2d\t", A[i]);

    printf("\nArray B... Not yet used numbers\n");
    i=0;
    for(i=0; i< Required_Num; i++)
    {
        if(B[i]!=0)
        { C[j]=B[i];
        printf("%2d\t", B[i]);
        j++;
        }

    }
    non_used=j;
    printf("\nInsert Sequence of "
           "Non-yet-used Numbers...\n");
    m=n=0;
    for(i=0; i<Required_Num; i++)
    {
        if(A[i]==0)
        {
            if( (j%2) == 0 )
            {
                A[i] = C[non_used-1-m]; m++;
            }
            else
            {
                A[i] = C[n]; n++;
            }
            printf("%2d\t", A[i]);
            j--;
        }
    }
    printf("\nAfter Modified...\n");
    for(i=0; i< Required_Num; i++)
        printf("%2d\t", A[i]);

    return 0;
}

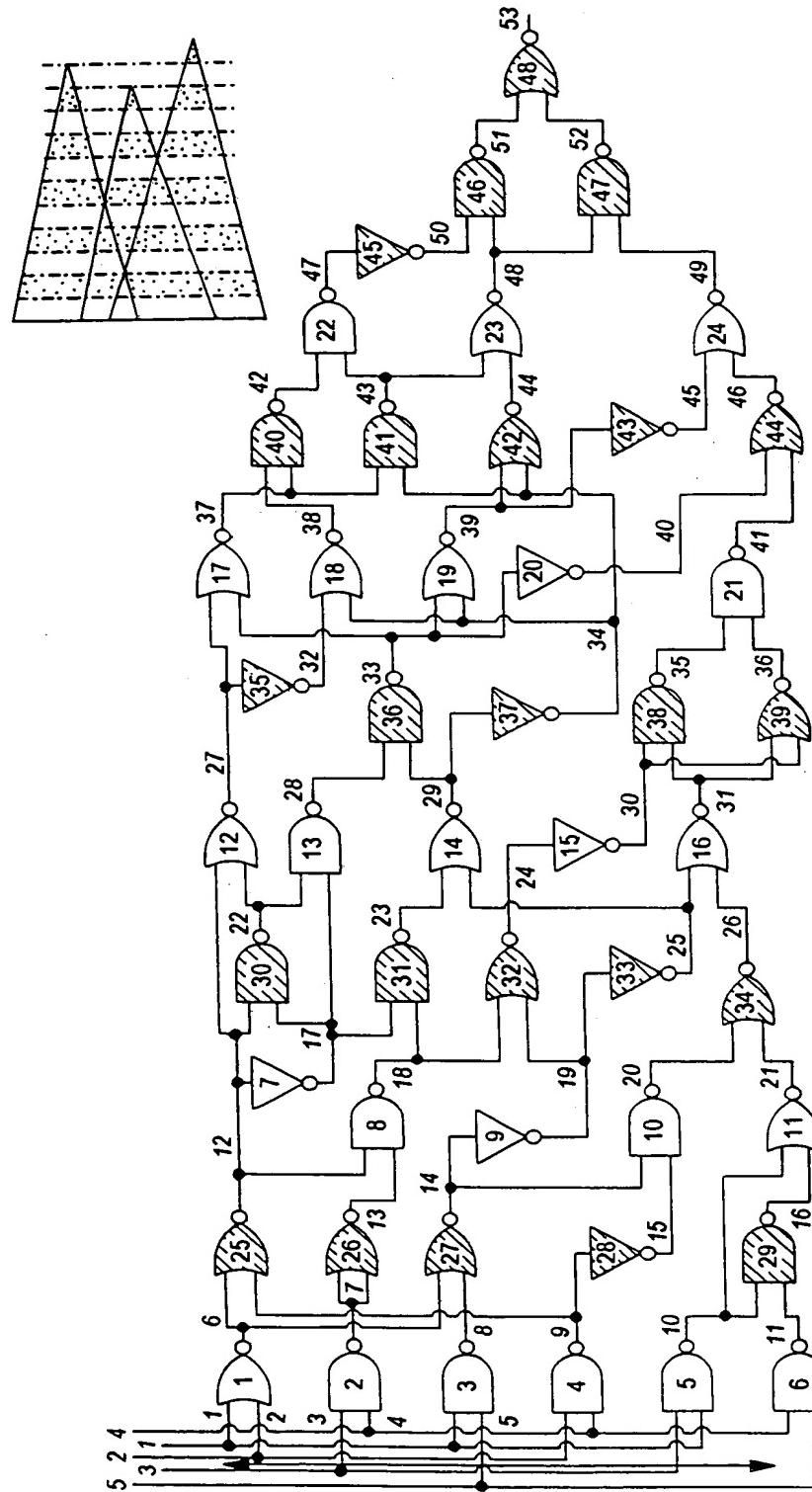
```

SOME OUTPUT RESULTS:														
Seed 35986, random numbers from 1 to 48														
38	45	42	5	31	44	47	4	22	23	9	36	27	7	32
6	11	19	6	13	9	41	3	29	11	43	23	32	16	1
2	26	47	30	42	17	28	29	2	26	47	30	42	17	28
Array A... Non-repeated generated numbers (from back-end):														
0	28	17	0	30	0	0	0	0	15	24	26	25	1	0
24	26	25	1	0	0	0	43	0	40	3	41	0	13	9
8	12	0	32	9	19	0	6	11	29	4	47	44	31	5
10	14	16	18	20	21	33	34	35	37	39	46	48	10	14
Array B... Not yet used numbers														
10	14	16	18	20	21	33	34	35	37	21	34	33	After Modified..	10
24	26	25	1	39	18	37	43	2	16	15	3	41	35	13
8	12	33	32	7	27	36	9	23	22	4	47	44	31	5
30	28	44	19	37	34	48	34	3	15	39	48	14	46	16
Insert Sequence of Non-yet-used Numbers...														
10	48	14	46	16	39	18	37	20	35	21	34	33	After Modified..	10
24	26	25	1	39	18	37	43	2	16	15	3	41	35	13
8	12	33	32	7	27	36	9	23	22	4	47	44	31	5
30	28	44	19	37	34	48	34	3	15	39	48	14	46	16
Seed 3350, random numbers from 1 to 48														
44	13	35	29	43	22	48	37	39	41	6	39	37	4	4
29	40	41	17	38	32	14	22	7	8	32	23	18	5	11
30	28	44	19	37	34	48	34	3	15	39	48	22	1	47
Array A... Non-repeated generated numbers (from back-end):														
0	0	0	19	0	28	30	0	47	1	26	11	5	0	18
0	14	32	0	17	0	40	0	27	15	38	31	46	0	4
37	48	22	43	29	35	13	44	6	41	39	33	36	42	45
Array B... Not yet used numbers														
2	3	9	10	12	16	20	21	24	25	45	2	42	3	36
10	14	32	25	17	12	40	24	27	15	38	31	46	16	4
37	48	22	43	29	35	13	44	6	41	39	33	36	42	45
Insert Sequence of Non-yet-used Numbers...														
24	16	21	20	36	9	33	10	25	12	45	2	42	3	36
After Modified..														
45	2	34	42	19	3	28	30	36	47	1	26	11	5	9
10	14	32	25	17	12	40	24	27	15	38	31	46	16	4
37	48	22	43	29	35	13	44	6	41	39	33	36	42	45

**FIG. 7**

# Replacement Sheet

APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	
DRAFTSMAN	



**FIG. 8A**

# Replacement Sheet

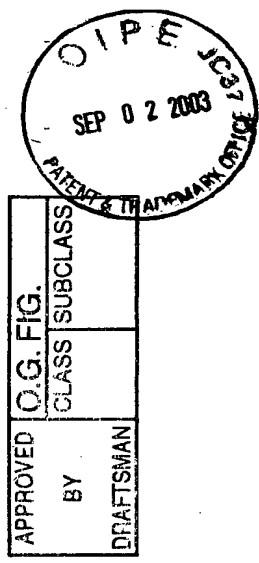
APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	
DRAFTSMAN	

Seed 34731, random numbers from 1 to 24											
1	10	21	8	17	6	4	7	22	15	21	10
9	9	12	13	12	19	6	4	10	21	15	7
23	11	4	24	12	12	19	6	4	10	21	15
13	12	0	9	15	22	7	4	6	17	19	0
8	21	10	1								
Array A... Non-repeated generated numbers (from back-end)											
24	0	11	23	0	0	0	19	0	17	6	4
13	12	0	9	15	22	7	4	6	17	19	0
8	21	10	1								
Array B... Not yet used numbers											
2	3	5	14	16	18	20					
Insert Sequence of Non-yet-used Numbers...	2	20	3	18	5	16	14				
After Modified...	24	2	11	23	20	3	18	5	19	6	16
	13	12	14	9	15	22	7	4	17	19	16
	8	21	10	1							
Seed 34797, random numbers from 25 to 48											
33	41	28	40	33	45	36	48	44	39		
27	47	35	37	30	31	44	33	46			
35	28	30	46								
0	0	25	46	0	25	46	0	31	30		
37	35	47	27	39	44	48	36	45			
40	28	41	33								
26	29	32	34	38	42	43					
Insert Sequence of Non-yet-used Numbers...	26	43	29	42	32	38	34				
After Modified...	26	43	29	42	27	42	32	38	34		
	37	35	47	27	39	44	48	36	45		
	40	28	41	33							

**FIG. 8B**



# Replacement Sheet



0. Initialize mapping (V, E) pairs to V-E plain, confirm the (V, E) pair distributed condition under nearly Max-cut reservation and may randomize the node number order.

1. Phase One: basic four steps.

( B ): Bottom-side base

( R ): Right-side base

( T ): Top-side base

( L ): Left-side base

2. Phase Two Begins: different additional steps can be choiced.

2A.	N	E	N	N	E	N		●	●	●
	(R)	(T)	(L)	(R)	(T)	(L)				

2B.	N	E	N	N	E	N		●	●	●
	(R)	(T)	(L)	(R)	(B)	(L)				

2C.	N	E	N	E	N	E	N	E	N	E
	(R)	(T)	(L)	(B)	(T)	(L)	(B)	(R)	(T)	(B)

N	E	N	E	N	E	N	N	E	N	E
(R)	(T)	(L)	(B)	(T)	(L)	(B)	(R)	(T)	(B)	(R)

2D.	E	N	E	N	E	N		●	●	●
	(B)	(R)	(T)	(L)	(B)	(R)	(T)	(L)		

2E. Some other recurring orders.

2F. Some other clustering techniques.

When every sort step completed, record nodes set, and if node set no more change, halt the procedures.

**FIG. 9**

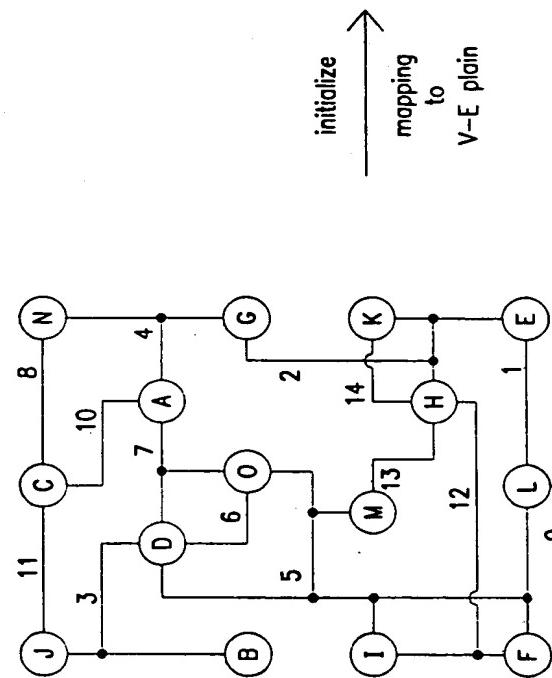
# Replacement Sheet



APPROVED	O. G. FIG.
BY	CLASS SUBCLAS
DRAFTSMAN	TRADEMARK OFFICE

	3	6	7	8	10	11
A	X	X	X	X	X	X
B	X	X	X	X	X	X
C	X	X	X	X	X	X
D	X	X	X	X	X	X
E	X	X	X	X	X	X
F	X	X	X	X	X	X
G	X	X	X	X	X	X
H	X	X	X	X	X	X
I	X	X	X	X	X	X
J	X	X	X	X	X	X
K	X	X	X	X	X	X
L	X	X	X	X	X	X
M	X	X	X	X	X	X
N	X	X	X	X	X	X
O	X	X	X	X	X	X

Confirm the distributed condition.



A 14 edges/15 nodes example.

**FIG. 10A**



## Replacement Sheet

APPROVED O.G. FIG.  
BY CLASS SUBCLASSEN  
DRAFTSMAN

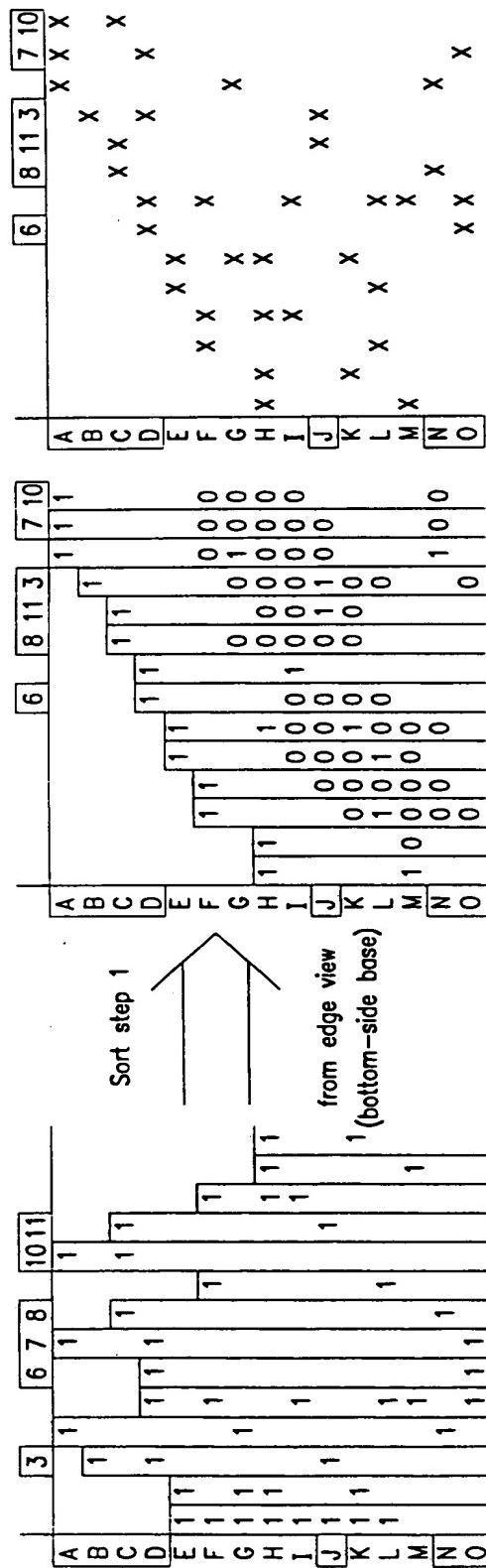
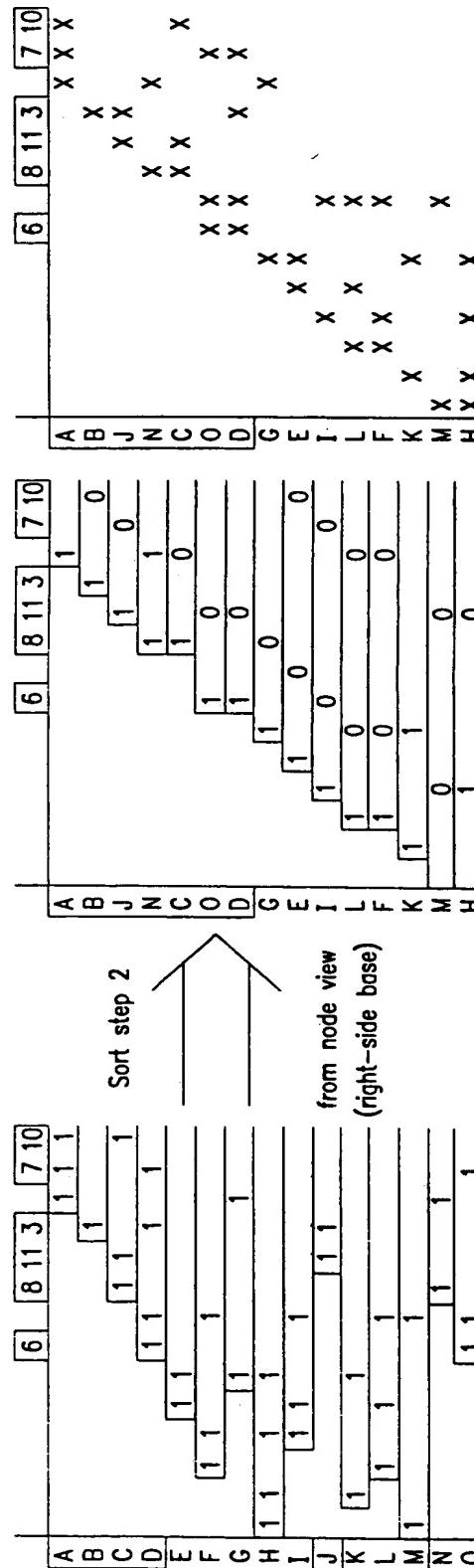


FIG. 10B

# Replacement Sheet



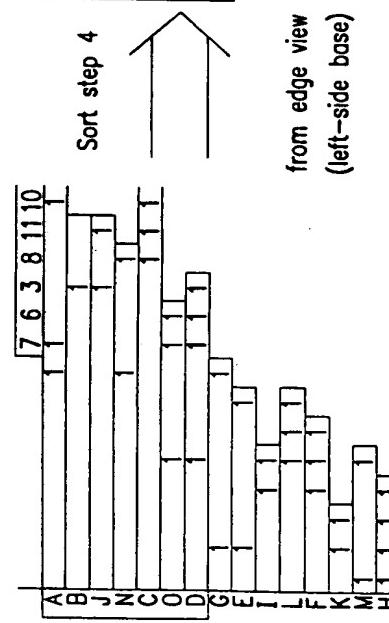
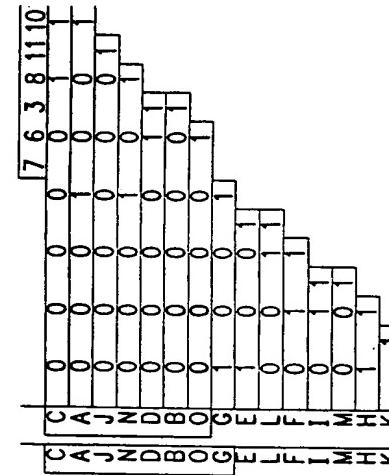
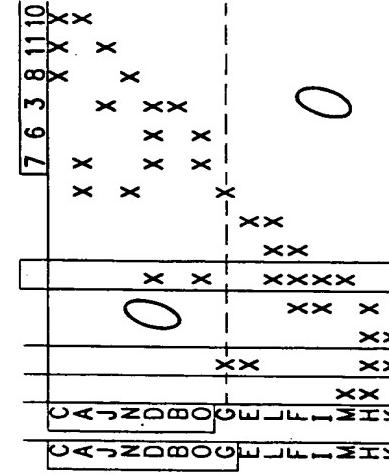
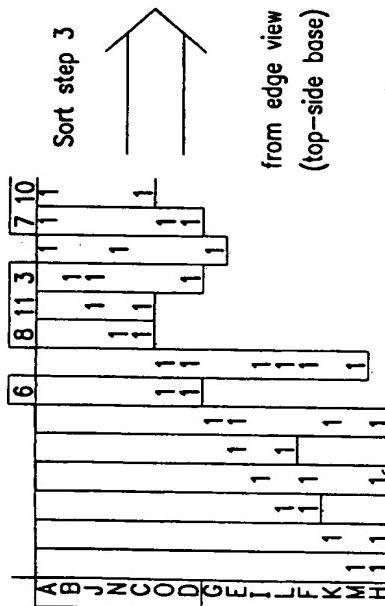
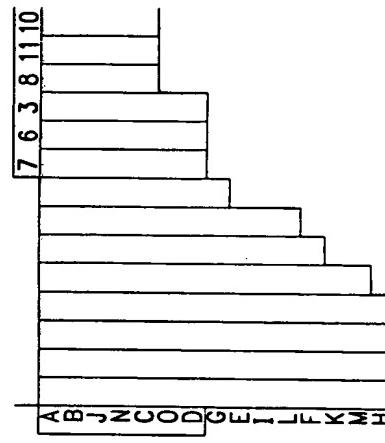
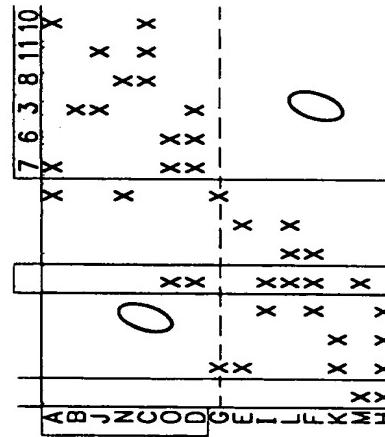
APPROVED	O.G. FIG.
BY	CLASS SUBCLASS
DRAFTSMAN	



# Replacement Sheet



APPROVED	O.G.	FIG.
CLASS	SUBCLASSE	TRADEMARK OFFICE
BY		
DRAFTSMAN		





## Replacement Sheet

APPROVED O.G. FIG.  
CLASS SUBCLASS  
BY DRAFTSMAN

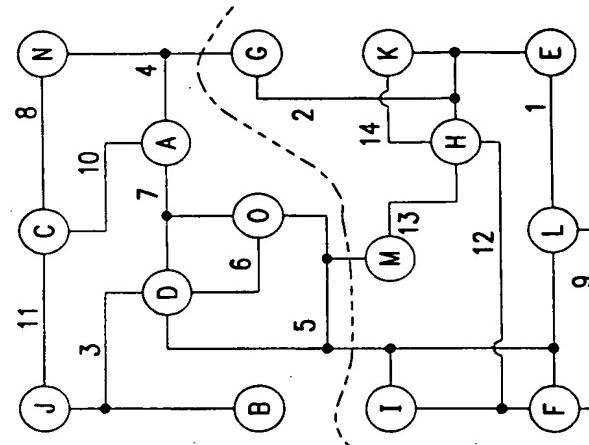
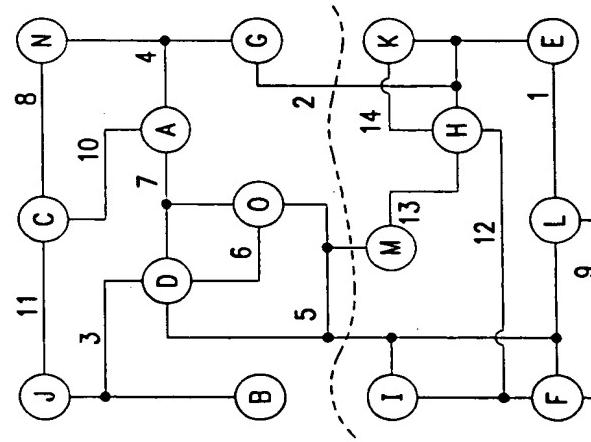


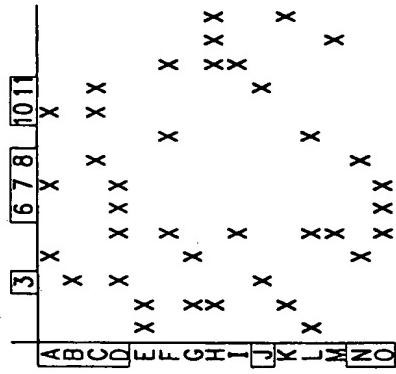
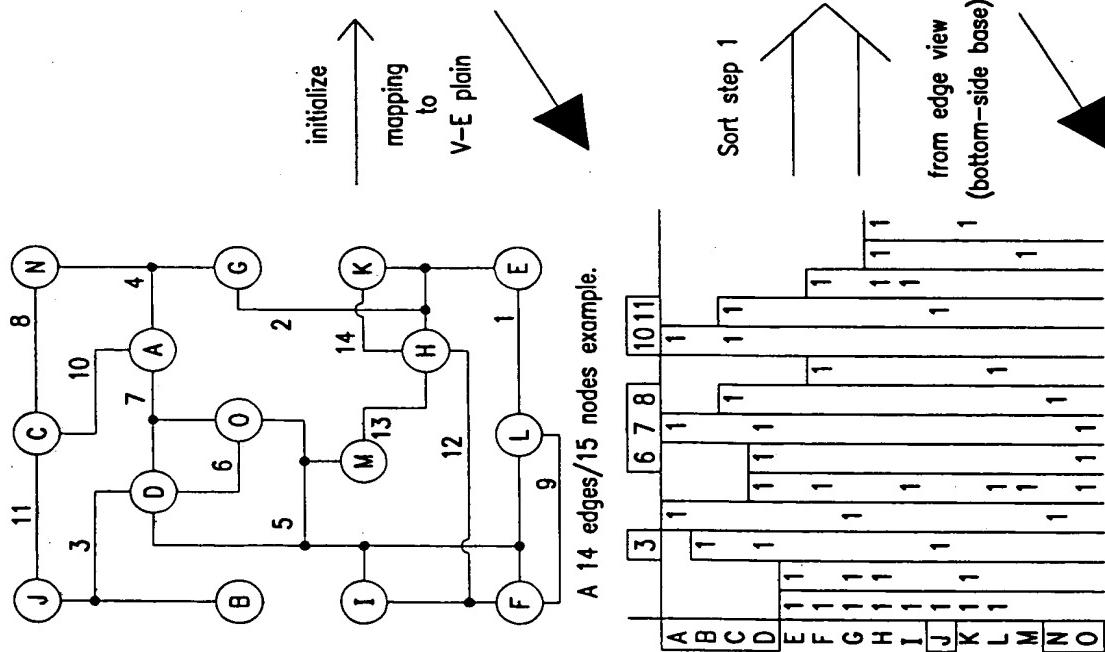
FIG. 10F



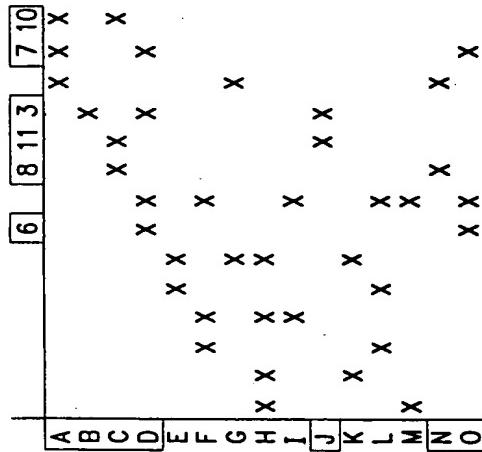


## Replacement Sheet

APPROVED	O.G. FIG.
BY	CLASS SUBCLASS
DRAFTSMAN	



Confirm the distributed condition.



**FIG. 11A**

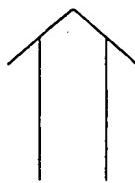
SEP 02 2003

APPROVED	O.G. FIG.
BY	CLASS SUBCLASS
DRAFTSMAN	

## Replacement Sheet

	6	8 11 3	7 10
A	B	J	N
C	D	O	G
E	F	I	L
G	H	K	M
J	L	M	H

Sort step 2

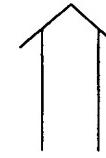


from node view  
(right-side base)



	6	8 11 3	7 10
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	

Sort step 3



from edge view  
(top-side base)



	6	8 11 3	7 10
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	

	7 6 3	8 11 10
X	X	X
X	X	X
X	X	X
O	XX	X
X	X	X
X	X	X
X	X	X
O	XX	X
X	X	X
X	X	X
X	X	X
H	X	X
H	X	X

FIG. 11B

Replacement Sheet

SEP 02 2003

APPLIED FOR	G. FIG.
BY	CLASS
DRAFTSMAN	SUBCLASS

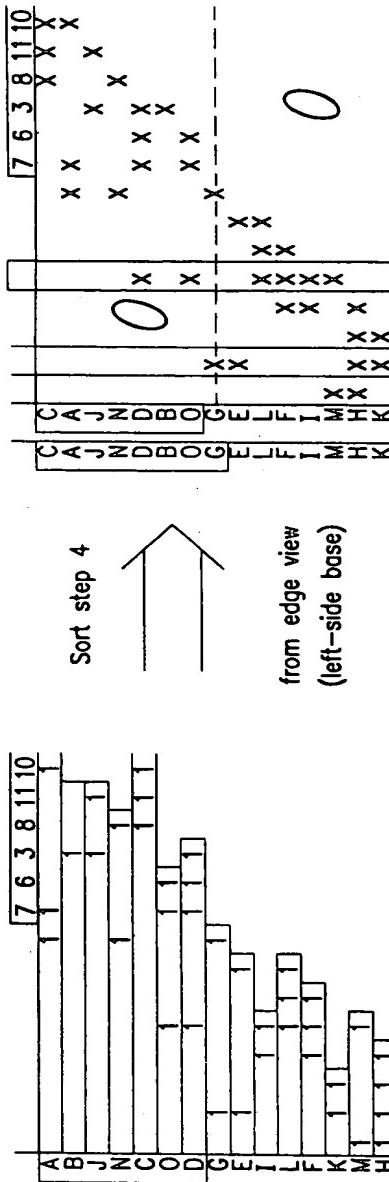
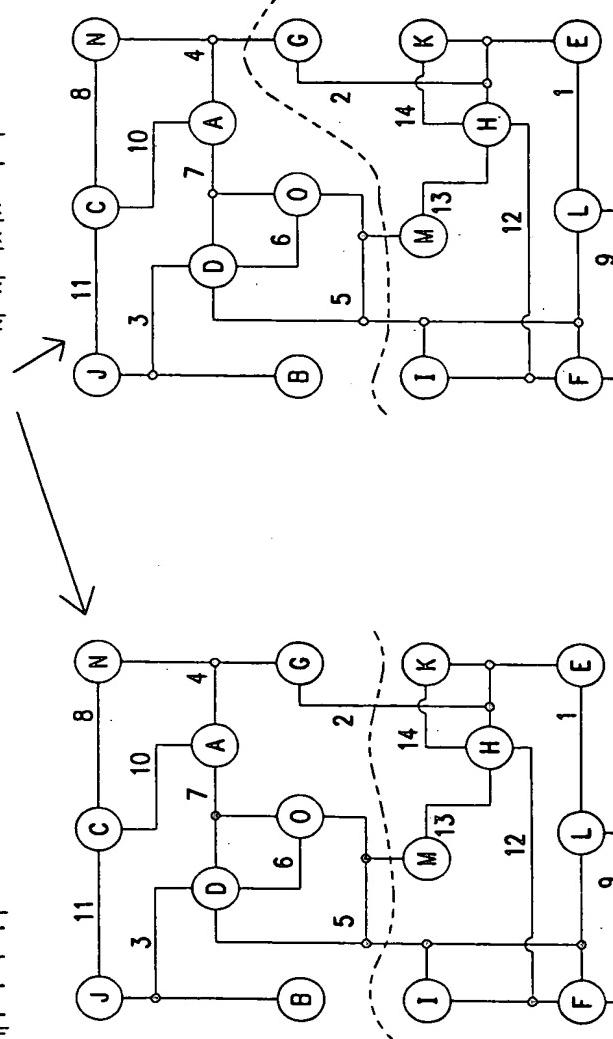
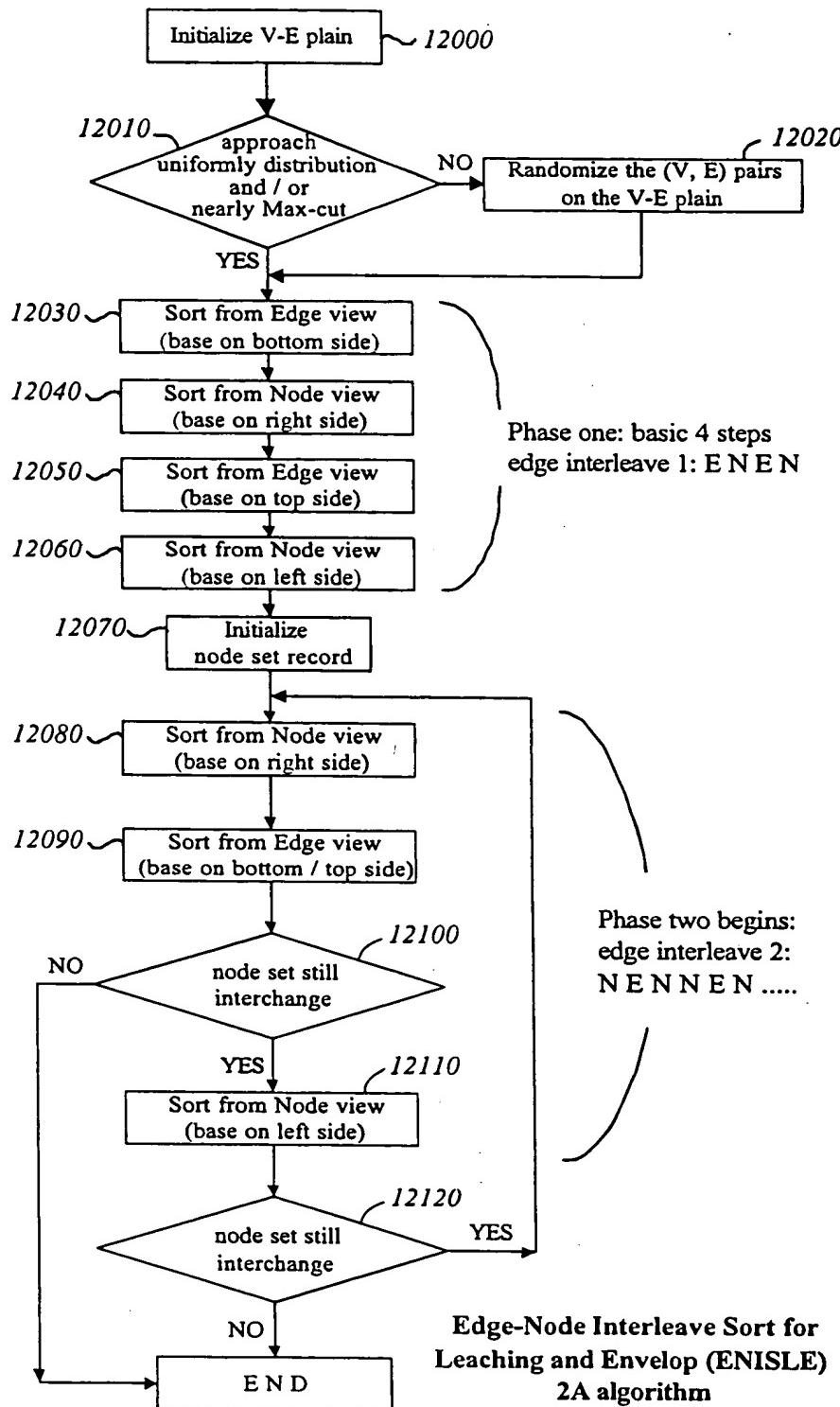
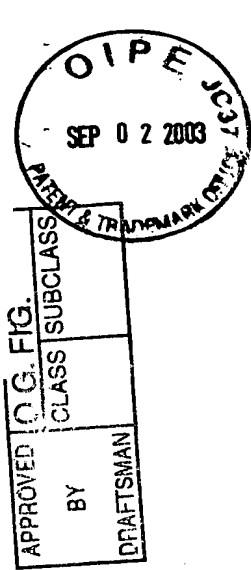


FIG. 11C



# Replacement Sheet

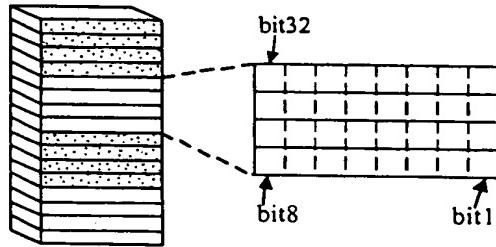


**FIG. 12**



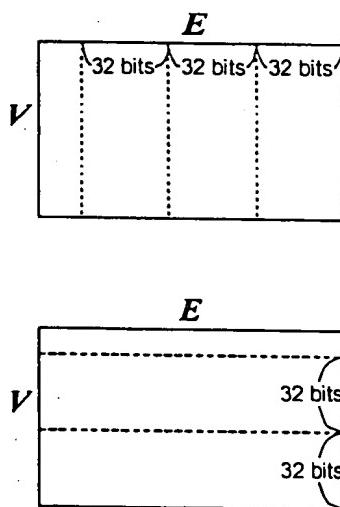
## Replacement Sheet

APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY DRAFTSMAN	



```
struct bitfield32 {
    bit32 :1;
    bit31 :1;
    bit30 :1;
    .....
    bit2 :1;
    bit1 :1;
} radix_sort_unit;
```

**FIG. 13**



### Radix Sorting (LSD) Example:

232, 321, 213, 231, 111, 112, 132, 123, 221  
 1S 321, 231, 111, 221  
 2S 232, 112, 132  
 3S 213, 123

321, 231, 111, 221, 232, 112, 132, 213, 123  
 10S 111, 112, 213  
 20S 321, 221, 123  
 30S 231, 232, 132

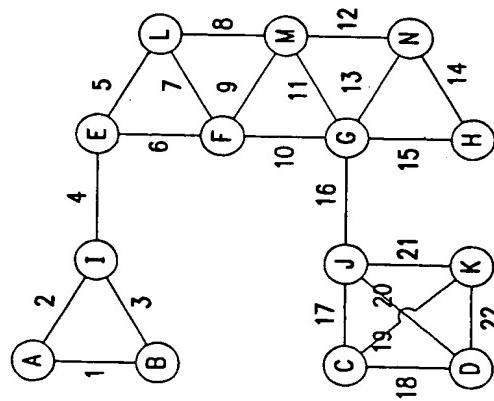
111, 112, 213, 321, 221, 123, 231, 232, 132  
 100S 111, 112, 123, 132  
 200S 213, 221, 231, 232  
 300S 321  
 Output: 111, 112, 123, 132, 213, 221, 231, 232, 321

**FIG. 14**



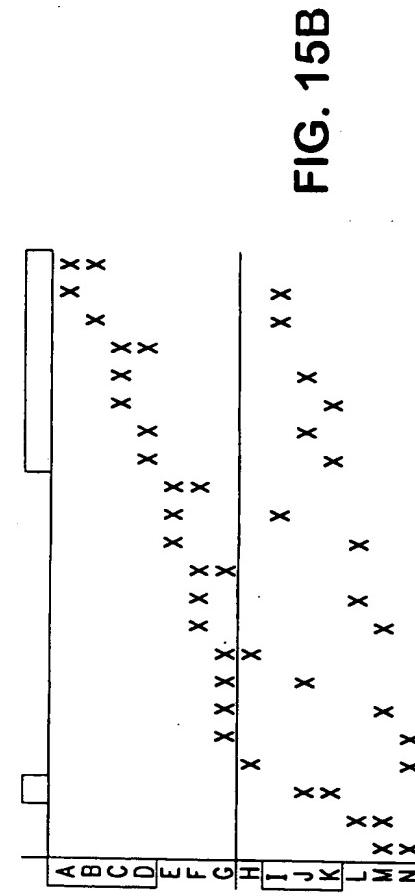
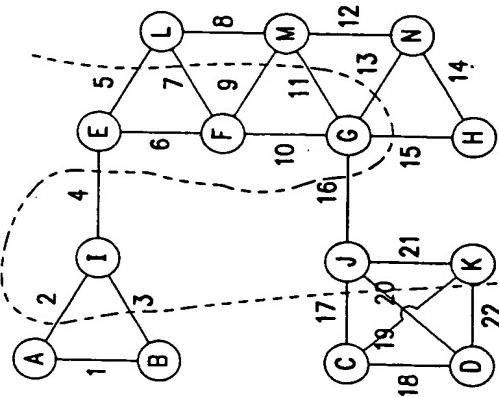
## Replacement Sheet

APPROVED	C.G. FIG.
BY	CLASS SUBCLASS
DRAFTSMAN	



**FIG. 15A**

Initialize the V-E Plain.



**FIG. 15B**

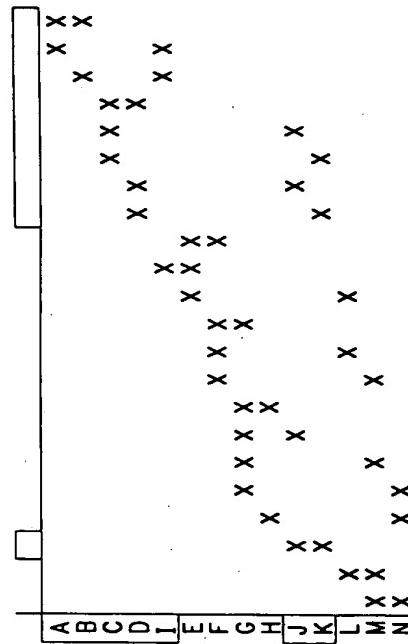
Step 1, cut numbers: 14,



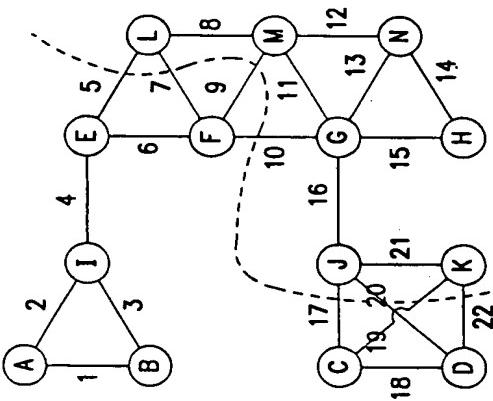
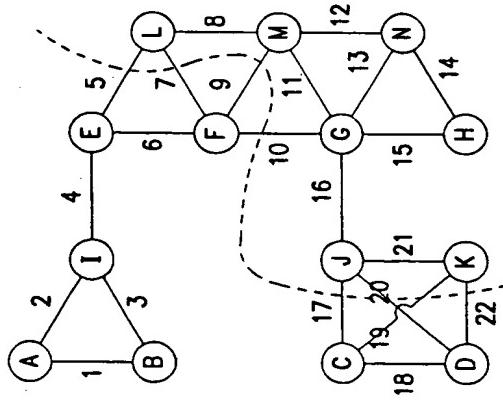
## Replacement Sheet

APPROVED	O.G. FIG.
BY	CLASS SUBCLAS
DRAFTSMAN	TRA

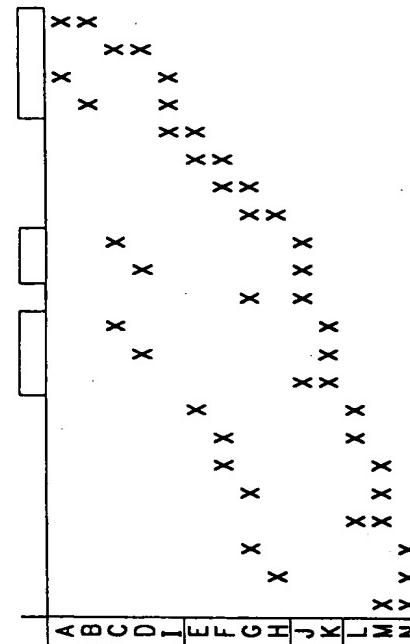
**FIG. 15C**



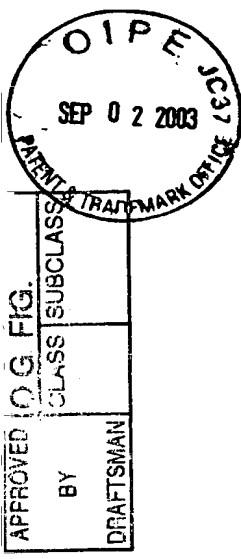
Step 2, cut numbers: 8.



**FIG. 15D**



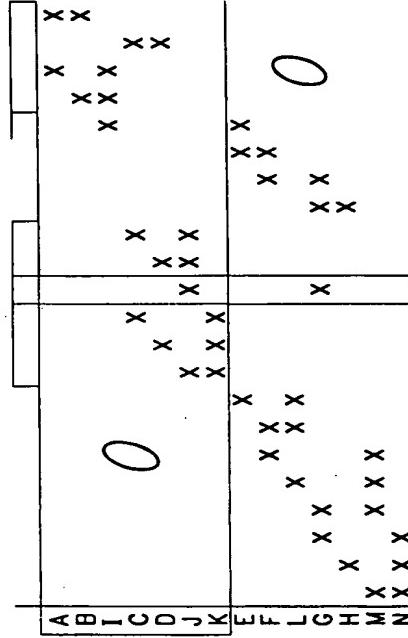
Step 3, 4, cut numbers: 8.



## Replacement Sheet

APPROVED	O G FIG
BY	CLASS SUB
	DRAFTSMAN

FIG. 15E



### **Step 5, cut numbers: 2.**

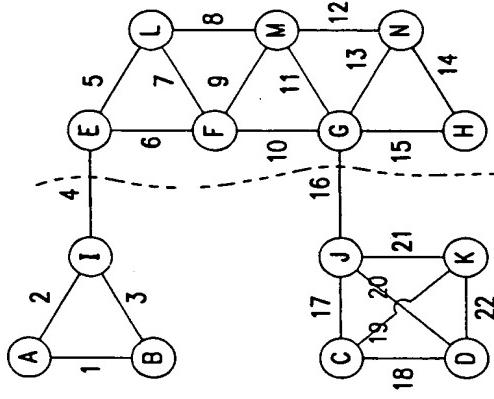
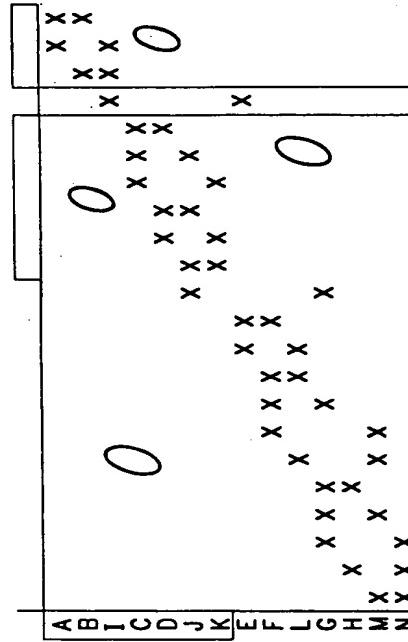
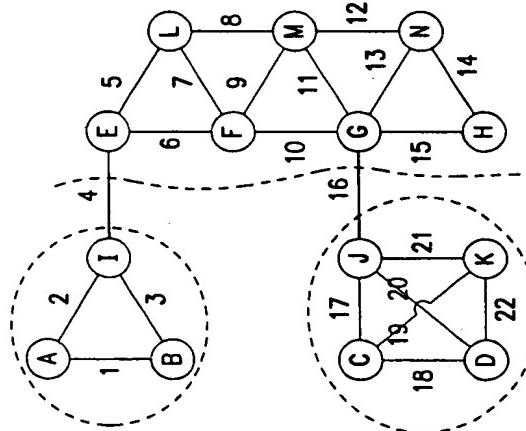


FIG. 15F



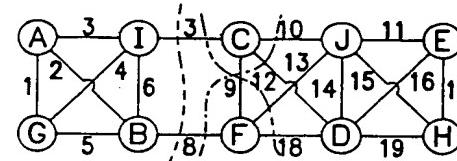
Step 6, cut numbers: 2.



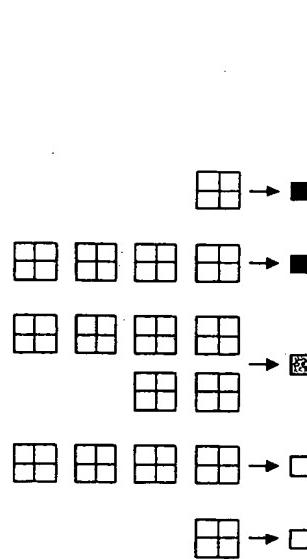
# Replacement Sheet

APPROVED	O.G. FIG.
CLASS / SUBCLAS	SSES
BY	DRAFTSMAN

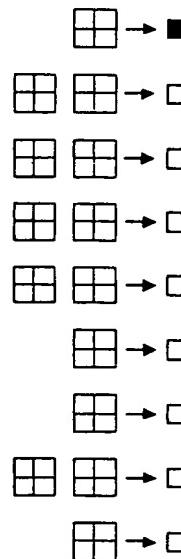
U.S. PATENT & TRADEMARK OFFICE  
SEP 02 2003  
C 37



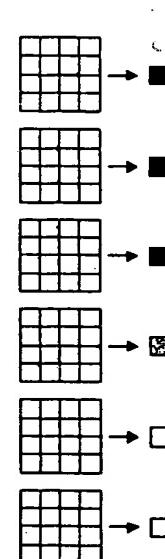
**FIG. 16**



**FIG. 17A**



**FIG. 17B**

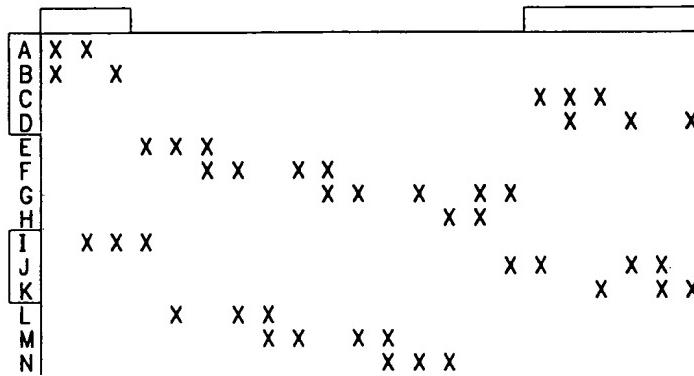


**FIG. 17C**

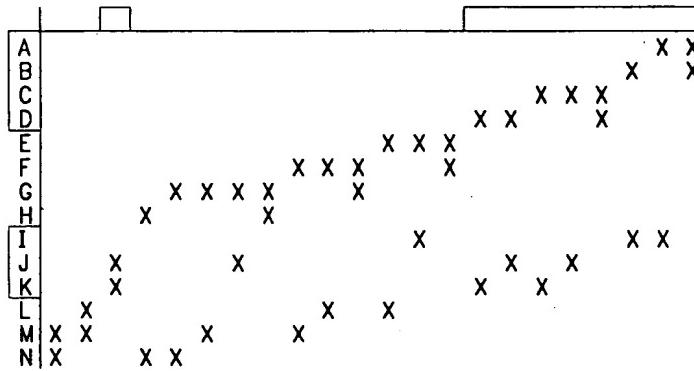


## Replacement Sheet

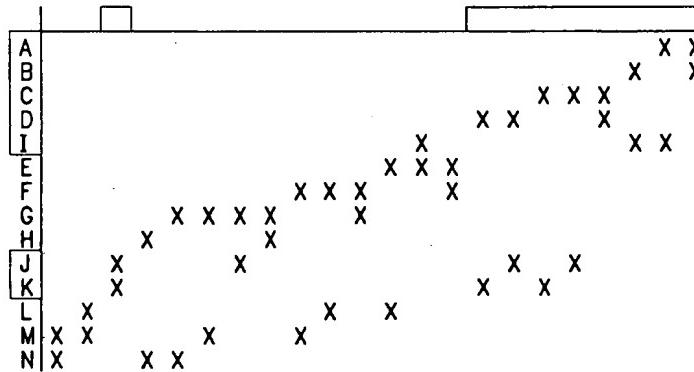
APPROVED	O.G. FIG.
CLASS	SUBCLASSES
BY	DRAFTSMAN



Initialize.



Step 1.



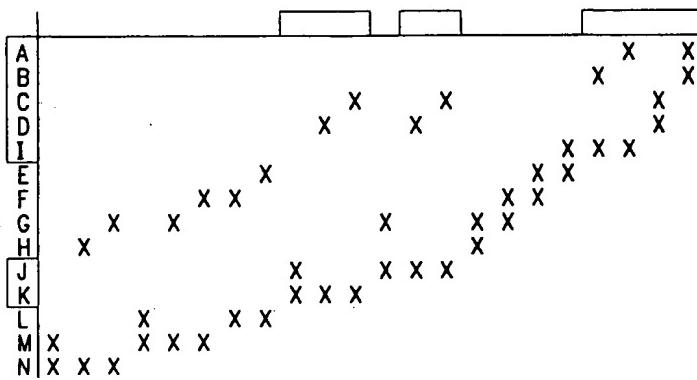
Step 2.

**FIG. 18A**

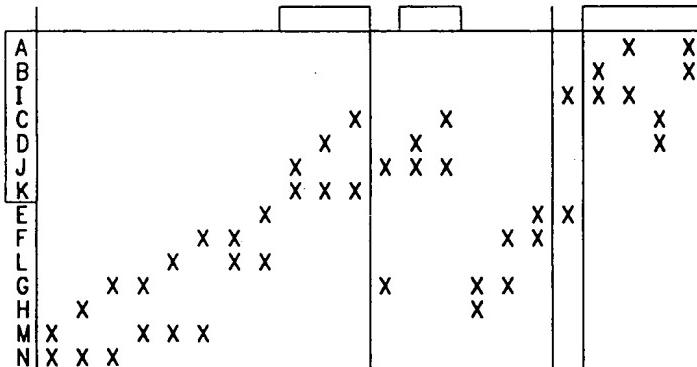


## Replacement Sheet

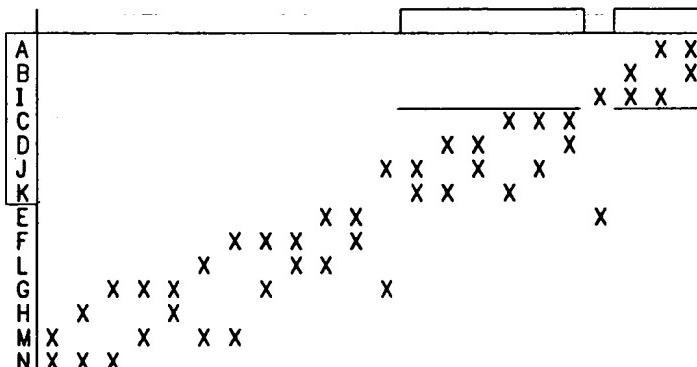
APPROVED O.G. FIG.  
CLASS SUBCLASSE  
BY DRAFTSMAN



### Step 3, 4.



## Step 5.



## **Step 6.**

FIG. 18B



## Replacement Sheet

APPROVED	G	FIG.
	CLASS	SUBCLASS
BY		
DRAFTSMAN		

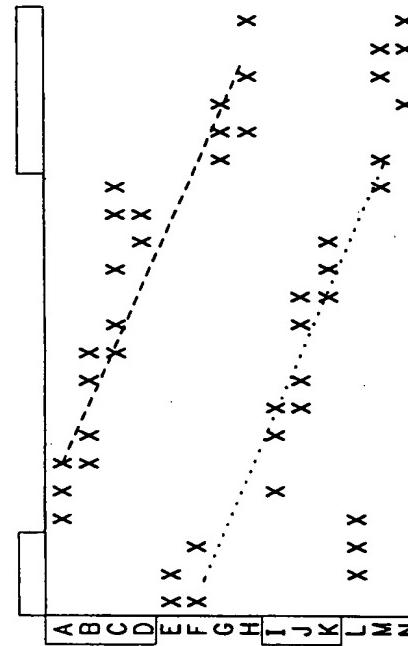
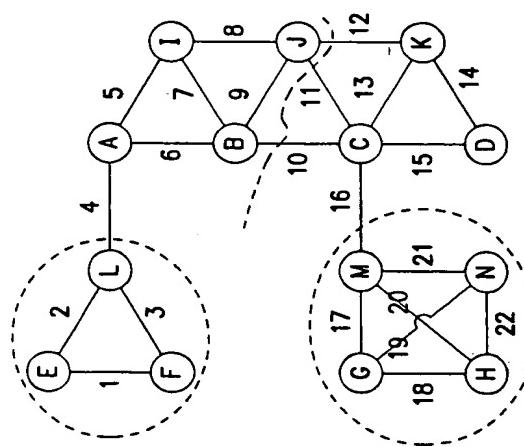


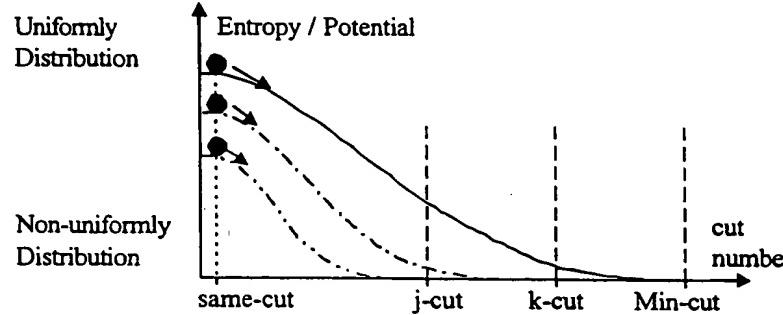
FIG. 19



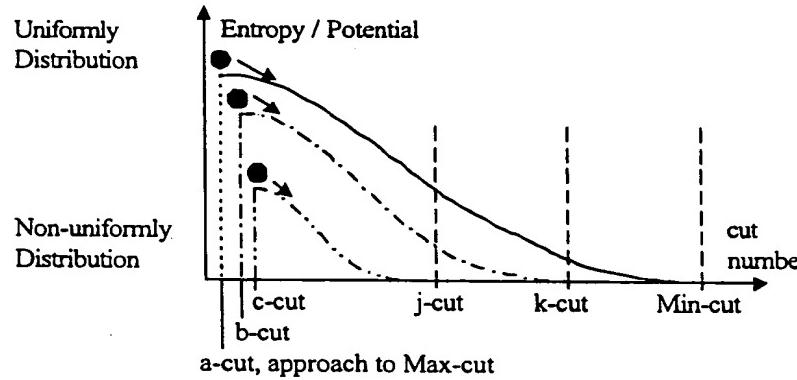


APPROVED	O.G. FIG.
CLASS	SUBCLASS
BY	DRAFTSMAN

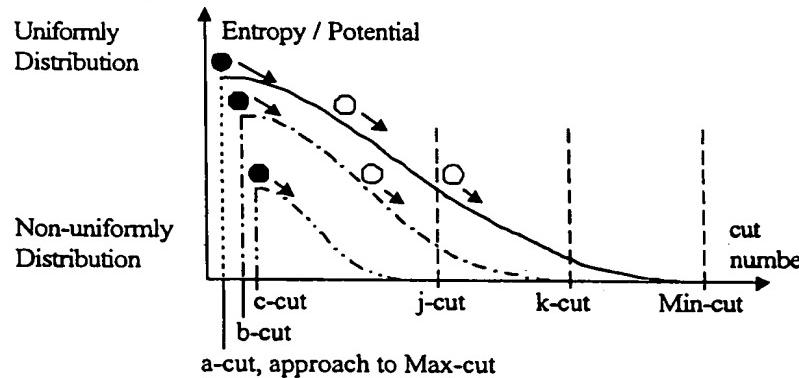
## Replacement Sheet



**FIG. 20A**



**FIG. 20B**



**FIG. 20C**